AMENDMENTS TO THE CLAIMS

1. (Currently amended) A computer system in which a virtual memory system is implemented, the computer system comprising a plurality of virtual memory pages, some of which being backed by a set of one or more physical memory pages and others of which being backed by a secondary memory, the computer system further comprising a primary data store containing a ~~first~~ primary data item and a ~~secondary~~ derived data store containing a ~~second~~ derived data item, the ~~first~~ primary data item and the ~~second~~ derived data item each providing memory mapping information about a first virtual memory page, the ~~second~~ derived data item being derived from the ~~first~~ primary data item, the computer system further comprising:

an actor, the actor modifying the ~~first~~ primary data item in the primary data store so that the ~~second~~ derived data item in the ~~secondary~~ derived data store is incoherent with the ~~first~~ primary data item;

a producer, the producer providing a first information indicating that the ~~first~~ primary data item in the primary data store has been modified; and

a consumer,

wherein the consumer receives the first information provided by the producer and, on occurrence of a coherency event at which the ~~secondary~~ derived data item is not to be incoherent with the primary data item, the consumer ~~receives the information provided by the producer and~~ eliminates the incoherency between the ~~second~~ derived data item and the ~~first~~ primary data item.

2. (New) The computer system of claim 1, wherein the producer also provides a second information indicating a derived value based on the modification that has been made to the primary data item and a third information that can be used to determine whether the second information remains valid, and the consumer uses the second information to modify the derived data item to make the derived data item coherent with the primary data item if the third information indicates that the second information is valid.

3. (New) The computer system of claim 2, wherein, on occurrence of the coherency event, if the third information indicates that the second information is no longer valid, the consumer eliminates the incoherency between the derived data item and the primary data item by invalidating the derived data item.

4. (New) The computer system of claim 2, wherein, on occurrence of the coherency event, if the third information indicates that the second information is no longer valid, the consumer determines a current value for the primary data item in the primary data store and uses this current value for the primary data item to modify the derived data item in the derived data store to make the derived data item coherent with the primary data item.

5. (New) The computer system of claim 2, wherein the second information and the third information comprise distinct data values.

6. (New) The computer system of claim 1, wherein the producer is the same software entity as the actor.

7. (New) The computer system of claim 1, wherein the coherency event comprises encountering a computer instruction that invalidates an entry within a Translation Lookaside Buffer.

8. (New) The computer system of claim 1, wherein the coherency event comprises a situation in which a Translation Lookaside Buffer is flushed.

9. (New) The computer system of claim 1, further comprising a patch channel that is used by the producer to communicate the first information to the consumer.

10. (New) The computer system of claim 9, wherein the producer can write to the patch channel at the same time that the consumer reads from the patch channel, without any data races.

11. (New) The computer system of claim 10, wherein the patch channel is stored in memory that is shared by the producer and the consumer, wherein an entry in the patch channel is indicated as being in-use by the producer at a point in time that is no earlier than the time when the rest of the entry is written to the patch channel, and wherein the entry in the patch channel is indicated as no longer being in-use by the consumer at a point in time that is no earlier than the time when the rest of the entry is read from the patch channel.

12. (New) The computer system of claim 9, wherein the patch channel comprises a half-duplex communication channel that is used exclusively by the producer and the consumer.

13. (New) The computer system of claim 12, wherein the patch channel also comprises a set data structure stored in memory that is shared by the producer and the consumer.

14. (New) The computer system of claim 13, wherein the set data structure comprises a data array containing a plurality of entries.

15. (New) The computer system of claim 9, further comprising additional producers and additional consumers, and wherein the patch channel is one of a plurality of patch channels in a patch channel matrix, wherein each patch channel in the patch channel matrix is used by a different pair consisting of a single producer and a single consumer.

16. (New) The computer system of claim 1, further comprising virtualization software for supporting a virtual machine, wherein the primary data store comprises a guest operating system page table containing translations from virtual memory pages to guest physical pages and the derived data store comprises a shadow page table containing translations from virtual memory pages to hardware physical pages.

17. (New) The computer system of claim 16, wherein the modification to the primary data item triggers a write trace that has been placed on the primary data store, and the producer provides the first information in response to the triggering of the write trace.

18. (New) The computer system of claim 16, wherein the computer system comprises a plurality of physical processors and the virtual machine comprises a plurality of virtual processors, wherein a first virtual processor is the producer and a second virtual processor is the consumer.

19. (New) The computer system of claim 16, wherein the computer system comprises a plurality of physical processors and the virtual machine comprises a plurality of virtual processors, wherein a first virtual processor is both the producer and the consumer.

20. (New) A computer program embodied in a tangible medium, the computer program being executable as part of a virtual computer system, the virtual computer system comprising physical hardware and virtualization software for supporting a virtual machine, the virtual machine running a guest operating system (OS) with a guest OS page table, the guest OS page table containing address translations from virtual addresses to guest physical addresses, the virtualization software generating a first shadow page table derived from entries in the guest OS page table, the first shadow page table containing address translations from virtual addresses to hardware physical addresses, the computer program eliminating incoherencies between the address

translations in the guest OS page table and the address translations in the first shadow page table, the guest OS page table containing a first primary entry providing an address translation for a first virtual address and the first shadow page table containing a first derived entry that was derived from the first primary entry in the guest OS page table, the first derived entry also providing an address translation for the first virtual address, the computer program comprising:

a first producer, the first producer responding to a first modification to the first primary entry in the guest OS page table, the first modification causing an incoherency between the first primary entry in the guest OS page table and the first derived entry in the first shadow page table, the first producer providing a first information indicating that a change has been made to a translation in the guest OS page table for the first virtual address; and

a first consumer, the first consumer receiving the first information from the first producer and, on occurrence of a coherency event at which the first shadow page table is not to be incoherent with the guest OS page table with respect to the first virtual address, the first consumer eliminates the incoherency between the first primary entry and the first derived entry.

21. (New) The computer program of claim 20, wherein the first producer also provides a second information indicating a derived value based on the first modification that has been made to the first primary entry and a third information that can be used to determine whether the second information remains valid, and the first consumer uses the second information to modify the first derived entry to make the first derived entry coherent with the first primary entry if the third information indicates that the second information is valid.

22. (New) The computer program of claim 20, wherein a patch channel is used exclusively by the first producer and the first consumer, the patch channel being used to communicate the first information from the first producer to the first consumer, and wherein an entry in the patch channel is indicated as being in-use by the first producer at a point in time that is no earlier than the time when the rest of the entry is written to the patch channel, and wherein the entry in the patch channel is indicated as no longer being in-use by the first consumer at a point in time that is no earlier than the time when the rest of the entry is read from the patch channel, so that the first producer can write to the patch channel at the same time that the first consumer reads from the patch channel, without any data races.

23. (New) The computer program of claim 20, further comprising a second consumer, wherein the virtualization software also generates a second shadow page table derived from entries in the guest OS page table, the second shadow page table also containing address translations from virtual addresses to hardware physical addresses, the computer program also eliminating incoherencies between the address translations in the guest OS page table and the address translations in the second shadow page table, the second shadow page table containing a second derived entry that also provides an address translation for the first virtual address, wherein the second derived entry was derived from the first primary entry in the guest OS page table prior to the first modification of the first primary entry so that the first modification to the first primary entry also causes an incoherency between the first primary entry and the second derived entry, wherein the virtual computer system further comprises a patch channel matrix comprising a first patch channel that is used exclusively by the first producer and the first consumer and a second patch channel that is used exclusively by the first producer and the second consumer, the first patch channel being used to communicate the first information from the first producer to the first consumer and the second patch channel being used to communicate the first information from the first producer to the second consumer, the second consumer also receiving the first information from the first producer and, on occurrence of a coherency event at which

the second shadow page table is not to be incoherent with the guest OS page table with respect to the first virtual address, the second consumer eliminates the incoherency between the first primary entry and the second derived entry.

24. (New) The computer program of claim 23, further comprising a second producer, the second producer responding to a second modification to a second primary entry in the guest OS page table, wherein the patch channel matrix further comprises a third patch channel that is used exclusively by the second producer and the first consumer and a fourth patch channel that is used exclusively by the second producer and the second consumer, the third patch channel being used to communicate a fourth information about the second modification from the second producer to the first consumer and the fourth patch channel being used to communicate the fourth information from the second producer to the second consumer.

25. (New) The computer program of claim 20, wherein the computer program is implemented within the virtualization software.

26. (New) A method for eliminating an incoherency between a primary data item in a primary data store and a derived data item in a derived data store, the primary data store and the derived data store being in a computer system, the derived data item being derived from the primary data item, the method comprising:

determining that a first modification is made to the primary data item in the primary data store, the first modification causing the derived data item in the derived data store to be incoherent with the primary data item;

providing a first information indicating that the primary data item in the primary data store has been modified; and

on occurrence of a coherency event at which the derived data item is not to be incoherent with the primary data item, and based on the first information, eliminating the incoherency between the derived data item and the primary data item.

27. (New) The method of claim 26, wherein the primary data item and the derived data item each provide memory mapping information about a first virtual memory page and the first information indicates that the primary data item has been modified by specifying a physical address of the primary data item.

28. (New) The method of claim 26, further comprising providing a second information indicating a derived value based on the first modification that has been made to the primary data item and a third information that can be used to determine whether the second information remains valid, and using the second information to modify the derived data item to make the derived data item coherent with the primary data item if the third information indicates that the second information is valid.

29. (New) The method of claim 26, wherein a producer provides the first information to a consumer using a patch channel that is used exclusively by the producer and the consumer, and the consumer eliminates the incoherency between the derived data item and the primary data item.

30. (New) The method of claim 29, wherein the consumer retrieves the first information provided by the producer and eliminates the incoherency between the derived data item and the primary data item both in response to the occurrence of the coherency event.

31. (New) The method of claim 29, wherein the incoherency between the derived data item and the primary data item is eliminated by invalidating the derived data item from the derived data store.

32. (New) The method of claim 29, wherein the incoherency between the derived data item and the primary data item is eliminated by modifying the derived data item to make the derived data item coherent with the primary data item.

33. (New) The method of claim 32, wherein, on occurrence of the coherency event, the consumer determines a current value for the primary data item in the primary data store and uses this current value for the primary data item to modify the derived data item in the derived data store to make the derived data item coherent with the primary data item.

34. (New) The method of claim 30, wherein the producer also provides a second information indicating a derived value based on the modification that has been made to the primary data item, and the consumer uses this second information to modify the derived data item to make the derived data item coherent with the primary data item.

35. (New) The method of claim 29, wherein the producer indicates that an entry in the patch channel is in-use at a point in time that is no earlier than the time when the rest of the entry is written to the patch channel, and wherein the consumer indicates that the entry in the patch channel is no longer in-use at a point in time that is no earlier than the time when the rest of the entry is read from the patch channel, so that the producer can write to the patch channel at the same time that the consumer reads from the patch channel, without any data races.

36. (New) A method for eliminating incoherencies between data items in one or more primary data stores and data items in one or more derived data stores in a computer system, the method comprising:

a first producer determining modifications that are made to data items in a first primary data store, the modifications to the data items in the first primary data store causing one or more incoherencies between the data items in the first primary data store and data items in a first derived data store and causing one or more incoherencies between the data items in the first primary data store and data items in a second derived data store, the data items in the first and second derived data stores being derived from the data items in the first primary data store;

the first producer providing information to a first consumer identifying data items in the first derived data store that are incoherent relative to the data items in the first primary data store;

the first producer providing information to a second consumer identifying data items in the second derived data store that are incoherent relative to the data items in the first primary data store;

on occurrence of a coherency event at which one or more data items in the first derived data store are not to be incoherent with corresponding data items in the first primary data store, the first consumer uses the information provided by the first producer to eliminate one or more incoherencies between the data items in the first derived data store and the data items in the first primary data store; and

on occurrence of a coherency event at which one or more data items in the second derived data store are not to be incoherent with corresponding data items in the first primary data store, the second consumer uses the information provided by the first producer to eliminate one or more incoherencies between the data items in the second derived data store and the data items in the first primary data store.

37. (New) The method of claim 36, further comprising:

a second producer determining modifications that are made to data items in a second primary data store, the modifications to the data items in the second primary data store causing one or more incoherencies between the data items in the second primary data store and data items in a third derived data store and causing one or more incoherencies between the data items in the second primary data store and data items in a fourth derived data store, the data items in the third and fourth derived data stores being derived from the data items in the second primary data store;

the second producer providing information to a third consumer identifying data items in the third derived data store that are incoherent relative to the data items in the second primary data store;

the second producer providing information to a fourth consumer identifying data items in the fourth derived data store that are incoherent relative to the data items in the second primary data store;

on occurrence of a coherency event at which one or more data items in the third derived data store are not to be incoherent with corresponding data items in the second primary data store, the third consumer uses the information provided by the second producer to eliminate one or more incoherencies between the data items in the third derived data store and the data items in the second primary data store; and

on occurrence of a coherency event at which one or more data items in the fourth derived data store are not to be incoherent with corresponding data items in the second primary data store, the fourth consumer uses the information provided by the second producer to eliminate one or more incoherencies between the data items in the fourth derived data store and the data items in the second primary data store.

38. (New) The method of claim 37, wherein the third derived data store is the same as the first derived data store, the third consumer is the same as the first consumer, the fourth derived data store is the same as the second derived data store, the fourth consumer is the same as the second consumer, and the second primary data store is the same as the first primary data store.

39. (New) The method of claim 38, wherein the computer system is a virtual computer system comprising a virtual machine, the virtual machine comprising a first virtual processor and a second virtual processor, wherein the first producer, the first consumer and the third consumer perform functions related to emulating the first virtual processor, and the second producer, the second consumer and the fourth consumer perform functions related to emulating the second virtual processor.

40. (New) The method of claim 36, further comprising the first producer providing additional information to the first consumer and to the second consumer indicating derived values based on the modifications that have been made to the data items in the first primary data store, the first consumer using the additional information to modify the data items in the first derived data store to eliminate one or more incoherencies between the data items in the first derived data store and the data items in the first primary data store, and the second consumer using the additional information to modify the data items in the second derived data store to eliminate one or more incoherencies between the data items in the second derived data store and the data items in the first primary data store.

41. (New) The method of claim 40, further comprising the first producer providing second additional information to the first consumer and to the second consumer that can be used to determine whether the derived values remain valid, the first consumer using the second additional information to determine which derived values remain valid and using the valid derived values to eliminate incoherencies between the data items in the first derived data store and the data items in the first primary data store, and the second consumer using the second additional information to determine which derived values remain valid and using the valid derived values to eliminate incoherencies between the data items in the second derived data store and the data items in the first primary data store.

42. (New) The method of claim 36, wherein the first producer provides information to the first consumer by writing the information to a first patch channel that is used exclusively by the first producer and the first consumer, the first producer writing the information to the first patch channel and the first consumer reading the information from the first patch channel so that the first producer and the first consumer can use the first patch channel at the same time, and wherein the first producer provides information to the second consumer by writing the information to a second patch channel that is used exclusively by the first producer and the second consumer, the first producer writing the information to the second patch channel and the second consumer reading the information from the second patch channel so that the first producer and the second consumer can use the second patch channel at the same time.

43. (New) A virtual computer system, the virtual computer system comprising a physical computer system with multiple physical processors, the physical computer system running a virtualization software for supporting a virtual machine with multiple virtual processors, the virtual machine running a guest operating system (OS) with a guest OS page table, the guest OS page table containing address translations from virtual addresses to guest physical addresses, the virtualization software generating a first shadow page table and a second shadow page table derived from entries in the guest OS page table, the first and second shadow page tables containing address translations from virtual addresses to hardware physical addresses, the first shadow page table being used by a first virtual processor and the second shadow page table being used by a second virtual processor, the guest OS page table containing a first primary entry providing an address translation for a first virtual address and the first shadow page table containing a first derived entry that was derived from the first primary entry in the guest OS page table, the first derived entry also providing an address translation for the first virtual address, the guest OS page table also containing a second primary entry providing an address translation for a second virtual address and the second shadow page table containing a second derived entry that was derived from the second primary entry in the guest OS page table, the second derived entry

also providing an address translation for the second virtual address, the virtual computer system comprising:

a producer;

a first consumer;

a second consumer; and

a patch channel matrix, the patch channel matrix comprising a first patch channel by which the producer communicates information to the first consumer and a second patch channel by which the producer communicates information to the second consumer,

wherein the producer responds to a first modification to the first primary entry in the guest OS page table, the first modification causing an incoherency between the first primary entry in the guest OS page table and the first derived entry in the first shadow page table, the producer writing a first information to the first patch channel indicating that a change has been made to a translation in the guest OS page table for the first virtual address,

wherein the producer responds to a second modification to the second primary entry in the guest OS page table, the second modification causing an incoherency between the second primary entry in the guest OS page table and the second derived entry in the second shadow page table, the producer writing a second information to the second patch channel indicating that a change has been made to a translation in the guest OS page table for the second virtual address,

wherein the first consumer reads the first information from the first patch channel and, on occurrence of a coherency event at which the first shadow page table is not to be incoherent with the guest OS page table with respect to the first virtual address, the first consumer eliminates the incoherency between the first primary entry and the first derived entry, and

wherein the second consumer reads the second information from the second patch channel and, on occurrence of a coherency event at which the second shadow page table is not to be incoherent with the guest OS page table with respect to the second virtual address, the second consumer eliminates the incoherency between the second primary entry and the second derived entry.

44. (New) The virtual computer system of claim 43, wherein the producer also provides to the first consumer a third information indicating a derived value based on the first modification that has been made to the first primary entry and a fourth information that can be used to determine whether the third information remains valid, and the first consumer uses the third information to modify the first derived entry to make the first derived entry coherent with the first primary entry if the fourth information indicates that the third information is valid.

45. (New) The virtual computer system of claim 43, wherein the first patch channel is used exclusively by the producer and the first consumer, and wherein an entry in the patch channel is indicated as being in-use by the producer at a point in time that is no earlier than the time when the rest of the entry is written to the patch channel, and wherein the entry in the patch channel is indicated as no longer being in-use by the first consumer at a point in time that is no earlier than the time when the rest of the entry is read from the patch channel, so that the producer can write to the first patch channel at the same time that the first consumer reads from the patch channel, without any data races.

46. (New) The virtual computer system of claim 43, wherein the first virtual address and the second virtual address are the same address and the first primary entry and the second primary entry are the same entry.

47. (New) The virtual computer system of claim 43, wherein a single shadow page table comprises both the first shadow page table and the second shadow page table.